1 Esercitazione Stata

18/11/2011 ore 9-12

Obiettivo: fornire rapidamente una introduzione di base a Stata; risolvere i problemi di inserimento e lettura dei dati.

Stata, che cos'è
Stata è un software statistico per l'elaborazione dei dati; produce statistiche, grafici e consente di stimare molti modelli econometrici. Per ulteriori informazioni si veda il sito www.stata.com
Per imparare ad usare Stata http://www.ats.ucla.edu/stat/stata/

come installare Stata
Inserire il cd di installazione e seguire le procedure automatiche. Vengono chieste due opzioni: la prima relativa alla scelta della cartella di lavoro (viene proposta la cartella con il nome c:\Stata); la seconda relativa alla versione delle componenti del programma (si consiglia la versione stata/SE, che è la più completa in alternativa la versione stata/intercooled). Per aprire il programma subito dopo l'installazione e necessario inserire il nome e i vari codici di accesso.

aprire e chiudere Stata
Per aprire il programma basta selezionare stata tra i programmi e cliccare oppure è possibile creare il collegamento al programma sul desktop.
Per chiudere stata si può scrivere exit nella finestra Stata Command oppure si clicca sul simbolo x come vale in generale per le finestre di windows.

come è fatto Stata
Il programma è composto da 4 finestre:

- STATA Command: é la finestra dove inserire i comandi
- STATA Results: é la finestra dei risultati
- Variables: é la finestra delle variabili (vengono visualizzate tutte le variabili presenti nel dataset caricato in memoria)
- Review: è la finestra su cui vengono visualizzati tutti i comandi digitati precedentemente attraverso la linea di comando.

In alto c'è una serie di menù e una barra degli strumenti che riproducono esattamente i comandi che vengono digitati nella finestra Stata command.

directory di lavoro
Per poter impostare la directory di lavoro è sufficiente digitare il comando nella finestra Stata Command:

```
cd c:\nomedir
```

per imparare a usare Stata `http://www.ats.ucla.edu/stat/stata/sk/default.htm`

Inserimento e lettura dei dati

Principali comandi

cd	Change directory
mkdir	Make a directory
dir	Show files in current directory
insheet	Read ASCII (text) data created by a spreadsheet
infile	Read unformatted ASCII (text) data
infix	Read ASCII (text) data in fixed format
import excel	Read data from a spreadsheet
describe	Describe contents of data in memory or on disk
compress	Compress data in memory
save	Store the dataset currently in memory on disk in Stata data format
use	Load a Stata-format dataset
count	Show the number of observations
list	List values of variables
clear	Clear the entire dataset and everything else

Sintassi di un comando Stata

```
command [varlist] [= exp] [weight] [if exp] [in range] [, options]
```

Per impostare la cartella di lavoro si utilizza il comando `cd`

```
cd c:\mydir
```

Per verificare quali file sono contenuti nella cartella si usa il comando `dir`

```
dir
```

Per creare una cartella di lavoro si utilizza il comando `mkdir`

```
mkdir c:\mydirnew
```

Per leggere dati in formato.csv (sia comma-separated sia tab-separated)si usa il comando `insheet`

```
insheet nomefile.csv, clear
```

N.B. l'opzione `clear` dopo la virgola consente di iniziare l'inserimento con l'editor dati vuoto, condizione necessaria per l'inserimento di nuovi dati. In alternativa si possono utilizzare i due comandi in sequenza

```
clear
insheet nomefile.csv
```

Se i dati in formato.csv non contengono come prima riga i nomi delle variabili si scrive l'elenco delle variabili nel comando

```
insheet var1 var2 var3 using nomefile.csv, clear
```

Per leggere dati in formato testo delimitato da spazi (generalmente salvati in formato .raw) si utilizza il comando `infile`

```
infile var1 var2 var3 using nomefile.raw, clear
```

Se i dati sono in formato ASCII, quindi in formato testo non separato da spazi, virgole o tabulazioni, è necessario utilizzare le tipiche informazioni relative al tracciato record. Un esempio:

Dati in formato ASCII data.fix

```
 195  094951
 26386161941
 38780081841
 479700  870
 56878163690
 66487182960
 786  069  0
 88194193921
 98979090781
107868180801
```

Tracciato record

variable name	column number
id	1-2
a1	3-4
t1	5-6
gender	7
a2	8-9
t2	10-11
tgender	12

È possibile inserire i dati utilizzando il comando **`infix`** e la seguente sintassi che prevede di indicare il nome della variabile e di seguito le colonne associate alla variabile

```
clear
infix id 1-2 a1 3-4 t1 5-6 gender 7 a2 8-9 t2 10-11 tgender 12 using data.fix
```

Per i dati in formato ASCII, quando i dataset dataset di grandi dimensioni si utilizza come supporto alla lettura un file dictionary che fornisce le istruzioni su come leggere correttamente le variabili. I file dictionary hanno estensione .dct e possono essere scritti con lo Stata editor (o quasiasi altro editor e salvati in formato .dct). Esempio di file dictionary

```
dictionary using data.txt

{

id          %2.0f "identificativo"
a1          %2.0f "var a1"
t1          %2.0f "var t1"
gender      %1.0f "genere"
a2          %2.0f "var a2"
t2          %2.0f "var t2"

}
```

Per dati in formato .xls si usa il comando **`import excel.`** Le opzioni (dopo la virgola) consentono di indicare il foglio di lavoro che contiene i dati e di segnalare che la prima riga contiene i nomi delle variabili

```
import excel data.xls", sheet("Sheet1") firstrow
```

N.B. tutti i comandi possono essere scritti nello Stata Command e mandati in esecuzione con il tasto invio oppure selezionati nel menù principale e, alcuni, tramite i tasti di scelta rapida (icone).

Dopo aver inserito I dati, il commando **`describe`** fornisce alcune informazioni sul dataset (numero osservazioni, numero variabili…).

Il comando **`compress`** riduce le dimensioni del dataset, laddove possibile.

Il comando **`save`** salva in dati inseriti nel formato dati di Stata: .dta.

```
save mydata.dta
```

Salva I dati nella cartella di lavoro impostata

```
save c:\mydir\mydata.dta
```

Salva i dati nella cartella c:\mydir
save mydata.dta, replace
salva i dati sovrascrivendoli

Per leggere dati che sono in formato .dta si usa il comando **use**
use schdat, clear
Se i dataset sono troppo grandi (per numero di osservazioni o variabili), il messaggio di errore segnala `no room to add more observation` (in rosso). Questo significa che è necessario allocare più memoria utilizzando il comando `set memory`
set memory 5m

Per conoscere il numero delle osservazioni si usa il comando **count**

Per visualizzare i valuti assunti dalle variabili si usa il comando **list** seguito dal nome delle varibili che si intende utilizzare
list id t1 gender

*** bla**
Stata non legge la riga preceduta da asterisco

/* bla … bla … bla … bla …
bla … bla … bla … bla … bla …*/
stata non legge le righe comprese tra barra-asterisco e asterisco-barra

command var // bla bla
il comando si interrompe dopo le due barre e stata legge la riga sotto

command var1 ///
var2
il comando continua nella riga sotto (per righe di comando lunghe)

Esplorazione dei dati

Principali comandi

use	Load dataset into memory
describe	Describe a dataset
list	List the contents of a dataset
codebook	Detailed contents of a dataset
summarize	Descriptive statistics
tabulate	One/two way frequency tables
tabstat	Table of descriptive statistics
table	Create a table of statistics
log	Create a log file
doedit	Open a do file

_____inizio file es1.do

```
* entering, managing, exploring and modifying data (y, BI)
* 18 nov 2011
/* dataset indagine sui bilanci delle famiglie della Banca d'Italia
www.bancaditalia.it in >statistiche e quindi in >indagini campionarie
scaricare e salvare i dataset in formato stata nella cartella c:\bi2008
scaricare anche i metadati e documneti utili nella stessa cartella */

clear                                // rimuove I dataset dalla memoria
set dp comma      // visualizzazione della virgola come separatore dei decimali
set more off                         //non blocca la schermata dei risultati
capture log close                    // chiude file .log eventualmente aperti
log using es1.log, replace           // apre un file .log sovrascrivendolo

cd c:/bi2008                         // cambia la dir di lavoro
*dir                                 // mostra I file contenuti nella dir

use carcom08                         // apre un file in formato stata
sort nquest nord                     // ordina in maniera crescente le variabili
save, replace

describe                             // descrive i contenuti del dataset

count                                // cont ail numero delle osservazioni

list in 1         // mostra tutti I valori delle var per la osservazione 1
list nquest nord eta area3 ireg                // " per le var in elenco
list nquest nord eta area3 ireg  in 1/10       // " per le osservazini da 1 a 10
*break button

use rfam08
sort nquest
save, replace

use rper08
sort nquest nord
save, replace
```

```
use carcom08
merge nquest using rfam08, keep(y) //unisce al dataset master (carcom08) i dati
del dataset using (rfam08) tramite la variabile chiave nquest

tabulate _merge         //fa la distribuzione di frequenza della var _merge
drop _merge             // elimina la variabile

rename y yfam           // rinomina la variabile y in yfam
label variable yfam "y disponibile netto famigliare"  // etichetta la variabile

sort nquest nord
merge nquest nord using rper08, keep(y yl ym yt yc) // l'opzione keep consente
di aggiungere al master dataset solo le variabili in elenco dello using dataset
tab _merge
drop _merge

label var y  "y disponibile netto individuale"
label var yl "y ind da lav dipendente"
label var ym "y ind da lav indipendente"
label var yt "y ind da trasferimenti"
label var yc "y ind da capitale"

summarize         // calcola alcune statistiche di base per tutte le var

summ eta          // " per la var in elenco
sum y
sum y, detail     // " in maniera più dettagliata
sum y [w=pesofit], det              // " aggiungendo il sistema dei pesi
sum y [w=pesofit] if y>0, det       // " e sotto la condizione espressa dopo if

graph box y if y>0 [w=pesofit]      // grafico box&whiskers
graph box y if y>0 & y<100000, ylabel(0(20000)100000) // " sotto le condiz. Dopo
if e con l'opzione per etichette asse delle y

histogram y, normal                 // istogramma con lla visualizzzaione della
normale
histogram y if y>0 & y<100000, normal

kdensity y if y>0 & y<100000, normal // funzione di densità Kernel

*DOMANDA: studiare le differenze nei redditi netti individuali per genere,
titolo di studio, area di residenza

drop if y<=0            // elimina tutte le var sotto la condizione dopo if
*keep if y>0            // mantiene tutte le var sotto la condizine dopo if

sum y [w=pesofit], det
sum y if sex==1 [w=pesofit], det
sum y if sex==2 [w=pesofit], det

*oppure
sort sex
by sex: summ y [w=pesofit]    // per ciscuna modalità della var dopo il by,
calcola le statistiche di base

*oppure
tabulate sex [w=pesofit], summarize(y)   // tabella di frequenza della var con
media e dev std

*oppure
tabstat y [w=pesofit], by(sex) // valore medio della var eta per le modalità di
sex
```

```
tabstat y eta [w=pesofit], by(sex) stat (mean p50 sd min max) // “ aggiunge
anche mediana, dev std nim e max
tabstat y eta [w=pesofit], stat (mean q sd range)


label variable sex "genere"
label value sex gen     // crea una etichetta per la var sex che si chiama gen
label define gen 1 "uomo" // definisce I valiti per l’etichetta gen
label define gen 2 "donna", add

graph box y if y<81918, over(sex) ylabel(0(25000)81918)

graph box y if y<81918, over(sex) over (area3) ylabel(0(25000)81918)
label value area3 rip3
label define rip3 1 "nord"
label define rip3 2 "centro", add
label define rip3 3 "sud+isole", add

graph box y if y<81918, over (area3) over(sex) ylabel(0(25000)81918)

tabulate studio

generate    studio3=1 if studio<=3  // genera una nuova variabile sotto
condizione
replace     studio3=2 if studio==4 | studio==5 // modifica i valori della nuova
variabile create
*replace    studio3=2 if studio>=4 & studio<=5
replace     studio3=3 if studio>5
*replace    studio3=3 if studio>=6

tab studio studio3
tab stud*

label value studio3 tit3
label define tit3 1 "tit basso"
label define tit3 2 "tit medio", add
label define tit3 3 "tit alto", add

graph box y if y<81918, over (studio3) over(sex) ylabel(0(25000)81918)
graph box y if y<81918, over (studio3) over(area3) ylabel(0(25000)81918)

table area3 studio3 sex, content (mean y) // crea una tabella con media della
variabile indicata tra parentesi dopo content
table area3 studio3 sex, content (mean y) col row  // “ dà anche i totali di
riga e di colonna
table area3 sex studio3, content (mean y) row
table area3 sex studio3, content (n  y) row // tabella con numero osservazioni
(n) anzichè la media

table area3 sex studio3 , content (mean y median y) row format(%6.0f) // “
tabella con valori medi e mediani, totali di riga e visualizzazione dei
risultati in max sei byte  di cui 0 decimali


log close         // chiude il file .log aperto
_____________________________________________________________fine file es1.do
```

Operatori (valgono le regole gerarchiche e si possono usare le parentesi)

Arithmetic		Logical		Relational (numeric and string)	
+	addition	&	and	>	greater than
-	subtraction	|	or	<	less than
*	multiplication	!	not	>=	> or equal
/	division	~	not	<=	< or equal
^	power			==	equal
-	negation			!=	not equal
+	string concatenation			~=	not equal

A double equal sign (==) is used for equality testing.

Generate & Replace

Hanno più o meno la stessa funzione – con replace Stata ti consente di non sovrascrivere accidentalmente una variabile esistente. Possono esistere più modi per generate una stessa variabile a seconda degli operatori che si utilizzano

```
generate    studio3=1 if studio<=3
replace     studio3=2 if studio==4 | studio==5
*replace    studio3=2 if studio>=4 & studio<=5
replace     studio3=3 if studio>5
*replace    studio3=3 if studio>=6
```

Recode

Per ottenere la variabile desiderata si può utilizzare recode, che modifica una variabile esistente o ne crea una nuova

```
recode sex (1=0) (2=1)
recode sex (1=0) (2=1), generate(donna)
```

rule	Example	Meaning
# = #	3 = 1	3 recoded to 1
# # = #	2 . = 9	2 and . recoded to 9
#/# = #	1/5 = 4	1 through 5 recoded to 4
nonmissing = #	nonmiss = 8	all other nonmissing to 8
missing = #	miss = 9	all other missings to 9